

**Ejercicio 1 (15pts)** Realice la gramática en notación EBNF para un bloque de manejo de excepciones en Java (lo ilustrado a continuación es sólo un ejemplo, considere la mayor cantidad de variantes tal como se vieron en la práctica. Recuerde que el lenguaje admite try/catch anidados):

```
try{
    sentencias
}
catch(Exception e){
    sentencias
}
finally{
    sentencias
}
```

b) (5) Realice el diagrama de Conway del ejercicio anterior.

**Ejercicio 2** Sea el siguiente código en C, indique para todos los identificadores indicando el número de línea

- ✓ a) (5) Su tipo de ligadura con el l-valor.
- ✓ b) (5) Su r-valor al momento de la compilación
- ✓ c) (7.5) Tiempo de vida y d) (7.5) Alcance.

#### PRIMER\_ARCHIVO.C

```
1.  int x;
2.  char *r;
3.
4.  main()
5.  {static int variable3;
6.   extern int a;
7.   int m, n;
8.   for(n=0; n<10; n++)
9.   { char var1='C';
10.    r=&var1;}
11. }
```

#### SEGUNDO\_ARCHIVO.C

```
12. static int auxiliar;
13. int a;
14. static int funcion2( )
15. { extern int x;
16.  auxiliar=auxiliar-2;
17.  ...
18. }
```

Realice este ejercicio sobre esta misma hoja.

Identif	L-value	R-Value	Alcance	T.V.
X	AUTOM	0	1-23	1-23
RT	AUTOM	NULL	2-23	1-23
MAIN	=	=	4-23	4-11
VARIABLE3	ESTATICA	0	5-23	1-23
2	AUTOM	BASURA	6-11	4-11
m	AUTOM	BASURA	7-11	4-11
n	AUTOM	BASURA	7-11	4-11
VAR1	AUTOM	BASURA	9-11	4-11
T	DINAM	BASURA	10-11	10-11
AUXILIAR	ESTATICA	0	12-23	1-23
2	AUTOM	BASURA	13-18	12-23
FUNCION2	-	=	14-23	14-18
X	AUTOM	BASURA	15-18	14-18

X AUXILIAR AUTOM BASURA 16-18 14-18

19.	int funcion3( )	FUNCION3	-	-	19-23	19-23
20.	{ int a;	2	AUTOM	BASICA	20-23	19-23
21.	a=a+4;					
22.	...					
23.	}					

Ejercicio 3 Responder V o F y justificar. Marque con un círculo la respuesta y justifique en hoja aparte.

- a) (7.5) Todos los lenguajes funcionales son fuertemente tipados. V **(F)**  
b) (7.5) Los parámetros formales por resultado pueden usarse en ejecución tal como se reciben en el procedimiento o función V **(F)**

Ejercicio 4 Sea el siguiente programa escrito en Pascal-like, realice la pila de ejecución,  
a) (20) Por cadena estática

Program Main;

Var z:integer; b: array [1..6] of integer;  
function a(x:integer);

```
begin
  if(x=1)then
    begin
      write("ultimo llamado");
      a:=x;
    end;
  else
    begin
      b[x]=b[x]+z;
      a:=a(x-1);
    end
  end
end
```

```
begin
  for z:=1 to 6 do begin
    b(z):= z;
    end;
  z:=a(3);
  for z:=1 to 6 do write (b(z));
end.
```

Nota: La forma de evaluación del lenguaje es de izquierda a derecha

Ejercicio 5 a)(10) Dado el siguiente código en Python. Describa los posibles caminos de ejecución.

```
#!/usr/bin/env python
#calc.py
```

```
def imprimir_posicion(x):
    print (("Resultado"), a[x]/a[x]-4)
```

0, 1, 2, 3, 4, 5

#La función range devuelve los números desde 0 al límite enviado como parámetro menos 1  
for x in range(6):

```
try:
    a = [0,1,2,3,4]
    imprimir_posicion(x)
except IndexError:
    print ("Ocurrió un error en el índice")
```

## Conceptos y Paradigmas de Lenguajes de Programación - 01/07/2022 Tema 1

Realice el parcial con TINTA (NO lápiz) - Presentismo con un ejercicio completo.

```
except ZeroDivisionError:
    print ("Ocurrió una división por cero")
else:
    print ("Se pudo acceder correctamente")
finally:
    print ("Vuelve a probar")
```

**b)(10p).** Indique cuáles son los tipos de datos identificados en el siguiente código C. Justifique en cada caso

```
#include <stdio.h>
#include <string.h>

struct Punto{
    int x;
    int y;
};

union estudiante
{
    char nombre[20];
    char apellido[20];
    float promedio;
};

int sumaPuntos(Punto p)
{
    int result;
    result = p.x + p.y;
    return result;
}
```



4-

CADENA ESTÁTICA

MAIN

\*1 RA MAIN

PR

EE

ED

Z = 1,2,3,4,5,6,3,1,2,3,4,5,6

b[1] = 1, 7 x

b[2] = 2, 8 v

b[3] = 3, 9 -

b[4] = 4

b[5] = 5

b[6] = 6

x()

VR

WRITE = 7, 8, 9, 4, 5, 6

\*2 RA 2

PR

EE \*1

ED \*1

x = 3

VR 2x

\*3 RA 2

PR

EE \*1

ED \*2

x = 2 v

VR 1 v

\*4 RA 2

PR

EE \*1

ED \*3

x = 1

VR

WRITE("ÚLTIMO LLAMADO")

main, JUNO QUE LOS CAMBIOS SE REFLEJAN

1)  $G = \{N, T, S, P\}$

$N = \{ \langle \text{EXCEPTION} \rangle, \langle \text{ID} \rangle, \langle \text{TRY} \rangle, \langle \text{CATCH} \rangle, \langle \text{SENTENCIA} \rangle, \langle \text{ID} \rangle, \langle \text{LETNUM} \rangle, \langle \text{LETRA} \rangle, \langle \text{FINALLY} \rangle \}$

$T = \{ \text{CATCH}, \{ \}, \text{TRY}, \text{EXCEPTION}(\text{ID}), \text{FOR}, \text{WHILE}, \text{FINALLY}, \dots, \text{ID}, \text{ID}, \text{ID}, \text{ID} \}$

$S = \{ P \}$

$P = \{ \langle \text{EXCEPTION} \rangle ::= \langle \text{TRY} \rangle \langle \text{CATCH} \rangle [ \langle \text{FINALLY} \rangle ]$

$\langle \text{TRY} \rangle ::= \text{TRY} \{ \langle \text{SENTENCIAS} \rangle \} + \{ \langle \text{CATCH} \rangle \} +$

$\langle \text{SENTENCIAS} \rangle ::= \{ \langle \text{SENTENCIA} \rangle \}^+ \{ \langle \text{TRY} \rangle \}^*$  esto debe estar dentro de SENTENCIAS

$\langle \text{CATCH} \rangle ::= \text{CATCH} (\text{EXCEPTION}(\text{ID})) [ \langle \text{FINALLY} \rangle ] \{ \langle \text{TRY} \rangle \}^*$

$\langle \text{SENTENCIA} \rangle ::= (\text{FOR}, \text{WHILE}, \dots)$  las opciones no se repiten por cons.

$\langle \text{ID} \rangle ::= \langle \text{LETRA} \rangle \{ \langle \text{LETRANUM} \rangle \}^*$

$\langle \text{LETRANUM} \rangle ::= (0 \dots 9 | a \dots z | A \dots Z)$  esta actividad produce un loop

$\langle \text{LETRA} \rangle ::= (a \dots z | A \dots Z)$

$\langle \text{FINALLY} \rangle ::= \text{FINALLY} \{ \langle \text{SENTENCIAS} \rangle \}$

- 5) UNO DE LOS POSIBLES CAMINOS SERIA: SE INGRESA AL FOR PERO AL IMPRIMIR LA POSICION ESTA NO EXISTIRIA EN EL ARREGLO, SERIA INVALIDA Y LA ATRAPARIA INDEXERROR, ESTO ASUMIENDO QUE NO SE DIVIDIO POR CERO COMO ES POR TERMINACION NO SE SIGUEN EJECUTANDO LINEAS SIGUIENTES Y SE PROCEDE A IMPRIMIR EL FINALLY. COMO EL TRY NO CONTIENE AL FOR ESTE CONTINUA CON LAS OTRAS POSICIONES.
- EN OTRO CASO SE PODRIA INTENTAR DIVIDIR POR CERO EN CUYO CASO LA EXCEPCION SE PROPAGA DINAMICAMENTE Y LLEVA A SER ATRAPADA POR ZERO DIVISIONERROR IMPRIMRIENDO "OCURRIO DIVISION X CERO" EL FINALLY TAMBIEN SE EJECUTA.
  - OTRO CAMINO SERIA UNA POSICION VALIDA Y UNA DIVISION VALIDA EN CUYO CASO SI SE PUEDE IMPRIMIR LA POSICION Y SE EJECUTA EL ELSE AL NO HABER LEVANTADO (SIGUE OTRO CAMINO) →